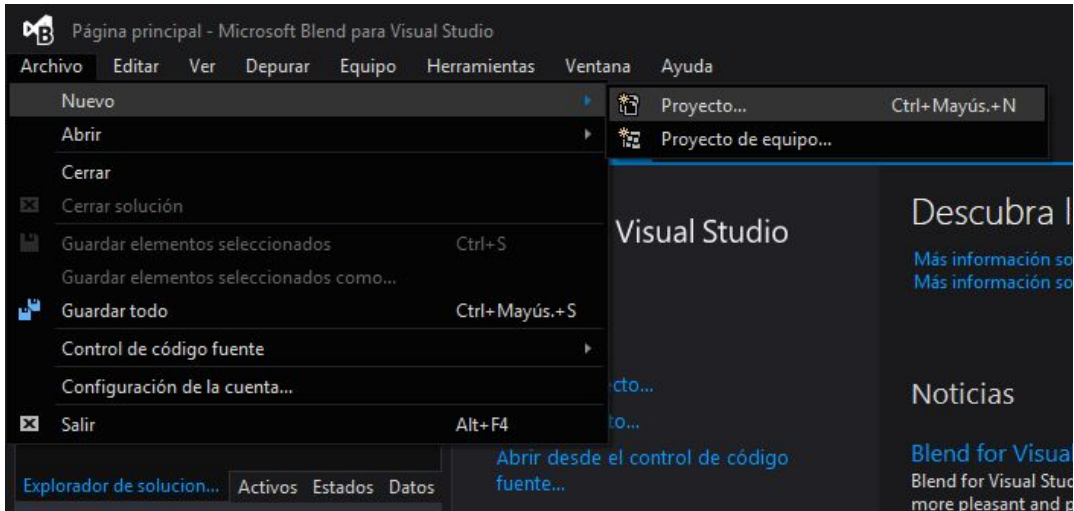


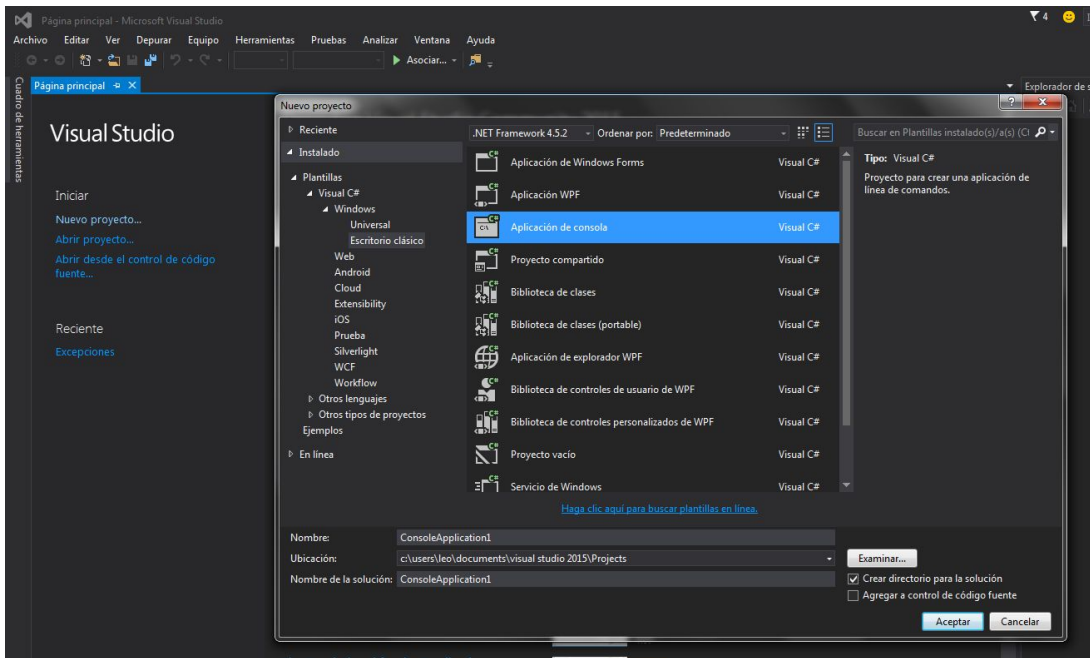
## Guía de ejercicios (1er parte)

### Realizar los siguientes ejercicios en proyectos de tipo consola.

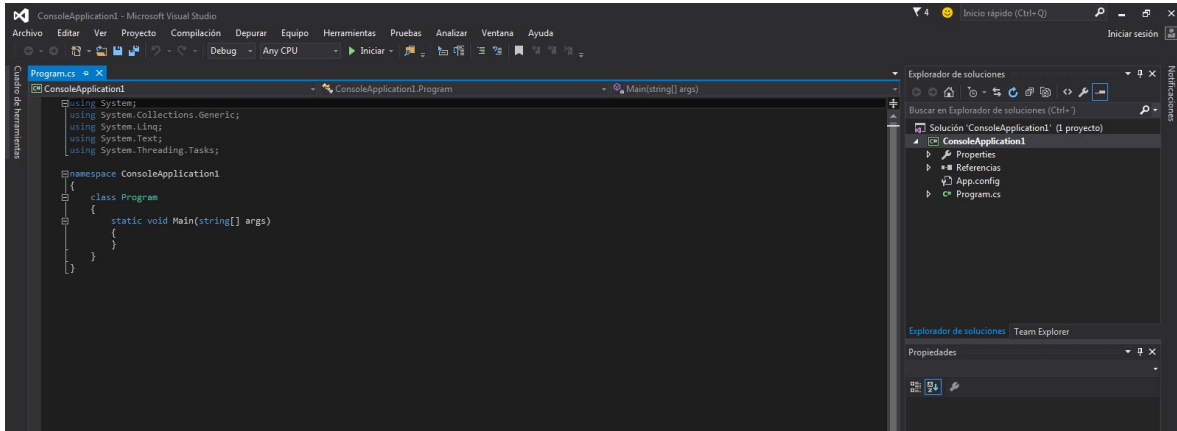
Vamos a Archivo->Nuevo->Proyecto



Luego seleccionamos Aplicación de Consola y elegimos un nombre



# A programar!



Referencia de MSDN: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>

## 1- Promedio

Escriba un programa que calcule el promedio de 4 notas ingresadas por el usuario:

Primera nota: **55**  
 Segunda nota: **71**  
 Tercera nota: **46**  
 Cuarta nota: **87**  
 El promedio es: 64.75

## 2- Conversión de unidades

- Escriba un programa que convierta de centímetros a pulgadas. Una pulgada es igual a 2.54 centímetros.

Ingrese longitud: **45**  
 45 cm = 17.7165 in  
 Ingrese longitud: **13**  
 13 cm = 5.1181 in

- Escriba un programa similar al anterior pero que convierta de pesos a dólares recibiendo la cotización a aplicar.

Ingrese valor en pesos: 100  
 Ingrese cotización del usd: 9.50  
 \$100 = u\$d950

## 3- Edad

Escriba un programa que entregue la edad del usuario a partir de su fecha de nacimiento:

## 4- Palabra más larga

Escriba un programa que pida al usuario dos palabras, y que indique cuál de ellas es la más larga y por cuántas letras lo es.

## 5- Letra o número

Escriba un programa que determine si un caracter ingresado es letra, número, o ninguno de los dos. En caso que sea letra, determine si es mayúscula o minúscula.

```
Ingrese caracter: 9
Es numero.
Ingrese caracter: A
Es letra mayúscula.
Ingrese caracter: f
Es letra minúscula.
Ingrese caracter: #
No es letra ni número.
```

**Tip:** Utilice los métodos estáticos ofrecidos por el type Char (<https://msdn.microsoft.com/es-es/library/system.char%28v=vs.110%29.aspx>)

## 6- Triángulos

Los tres lados  $a$ ,  $b$  y  $c$  de un triángulo deben satisfacer la [desigualdad triangular](#): cada uno de los lados no puede ser más largo que la suma de los otros dos.

$$a < (b + c), \quad b < (a + c), \quad c < (a + b)$$

Escriba un programa que reciba como entrada los tres lados de un triángulo, e indique:

- si acaso el triángulo es inválido; y
- si no lo es, qué tipo de triángulo es.

```
Ingrese a: 3.9
Ingrese b: 6.0
Ingrese c: 1.2
No es un triangulo valido.
```

```
Ingrese a: 1.9
Ingrese b: 2
Ingrese c: 2
El triangulo es isoceles.
```

```
Ingrese a: 3.0
Ingrese b: 5.0
Ingrese c: 4.0
El triangulo es escaleno.
```

## 7- Suma entre números

Escriba un programa que pida al usuario dos números enteros, y luego entregue la suma de todos los números que están entre ellos. Por ejemplo, si los números son 1 y 7, debe entregar como resultado  $2 + 3 + 4 + 5 + 6 = 20$ .

```
Ingrese num: 1
Ingrese num: 7
La suma es 20
```

## 8- Años bisiestos

Cuando la Tierra completa una órbita alrededor del Sol, no han transcurrido exactamente 365 rotaciones sobre sí misma, sino un poco más. Más precisamente, la diferencia es de más o menos un cuarto de día.

Para evitar que las estaciones se desfasen con el calendario, el calendario juliano introdujo la regla de introducir un día adicional en los años divisibles por 4 (llamados [bisiestos](#)), para tomar en consideración los cuatro cuartos de día acumulados.

Sin embargo, bajo esta regla sigue habiendo un desfase, que es de aproximadamente  $3/400$  de día.

Para corregir este desfase, en el año 1582 el papa Gregorio XIII introdujo un nuevo calendario, en el que el último año de cada siglo dejaba de ser bisiesto, a no ser que fuera divisible por 400.

Escriba un programa que indique si un año es bisiesto o no, teniendo en cuenta cuál era el calendario vigente en ese año:

```
Ingrese un año: 1988
1988 es bisiesto
Ingrese un año: 2011
2011 no es bisiesto
Ingrese un año: 1700
1700 no es bisiesto
Ingrese un año: 1500
1500 es bisiesto
Ingrese un año: 2400
2400 es bisiesto
```

## 9- Más corta y más larga

Desarrolle un programa que tenga la siguiente entrada:

- primero, el usuario ingresa un número entero  $n$ , que indica cuántas palabras ingresará a continuación;
- después el usuario ingresa  $n$  palabras.

La salida del programa debe mostrar la palabra más larga y la más corta que fueron ingresadas por el usuario.

La ejecución del programa debe verse así:

Cantidad de palabras: 5  
 Palabra 1: **negro**  
 Palabra 2: **amarillo**  
 Palabra 3: **naranja**  
 Palabra 4: **azul**  
 Palabra 5: **blanco**  
 La palabra mas larga es amarillo  
 La palabra mas corta es azul

## 10- Índice de masa corporal

Ejercicio sacado de [\[Camp09\]](#).

El riesgo de que una persona sufra enfermedades coronarias depende de su edad y su índice de masa corporal:

	edad < 45	edad ≥ 45
IMC < 22.0	bajo	medio
IMC ≥ 22.0	medio	alto

El índice de masa corporal es el coeficiente entre el peso del individuo en kilos y el cuadrado de su estatura en metros.

Escriba un programa que reciba como entrada la estatura, el peso y la edad de una persona, y le entregue su condición de riesgo.

## 11- Dígitos

Escriba un programa que determine la cantidad de dígitos en un número entero ingresado por el usuario:

Ingrese numero: **2048**  
 2048 tiene 4 digitos  
 Ingrese numero: **12**  
 12 tiene 2 digitos  
 Ingrese numero: **0**  
 0 tiene 1 digito

## 12- Caballo de ajedrez

Un tablero de ajedrez es una grilla de  $8 \times 8$  casillas. Cada celda puede ser representada mediante las coordenadas de su fila y su columna, numeradas desde 1 hasta 8.

El [caballo](#) es una pieza que se desplaza en forma de L: su movimiento consiste en avanzar dos casillas en una dirección y luego una casilla en una dirección perpendicular a la primera:

Escriba un programa que reciba como entrada las coordenadas en que se encuentra un caballo, y entregue como salida todas las casillas hacia las cuales el caballo puede desplazarse.

Todas las coordenadas mostradas deben estar dentro del tablero.

Si la coordenada ingresada por el usuario es inválida, el programa debe indicarlo.

```
Ingrese coordenadas del caballo.
Fila: 2
Columna: 8
El caballo puede saltar de 2 8 a:
1 6
3 6
4 7
```

```
Ingrese coordenadas del caballo.
Fila: 3
Columna: 4
El caballo puede saltar de 3 4 a:
1 3
1 5
2 2
2 6
4 2
4 6
5 3
5 5
```

```
Ingrese coordenadas del caballo.
Fila: 1
Columna: 9
Posicion invalida.
```

## 13- Números primos

Un [número primo](#) es un número natural que sólo es divisible por 1 y por sí mismo.

Los números que tienen más de un divisor se llaman números compuestos. El número 1 no es ni primo ni compuesto.

1. Escriba un programa que reciba como entrada un número natural, e indique si es primo o compuesto:
2. Ingrese un numero: **17**  
17 es primo  
Ingrese un numero: **221**  
221 es compuesto
3. Escriba un programa que muestre los  $n$  primeros números primos, donde  $n$  es ingresado por el usuario:
  - o Cuantos primos: **10**
  - o 2

- ☐ 3
- ☐ 5
- ☐ 7
- ☐ 11
- ☐ 13
- ☐ 17
- ☐ 19
- ☐ 23

4. Escriba un programa que muestre los números primos menores que  $m$ , donde  $m$  es ingresado por el usuario:

```
Primos menores que: 19
2
3
5
7
11
13
17
```

5. Escriba un programa que cuente cuántos son los números primos menores que  $m$ , donde  $m$  es ingresado por el usuario:

6. Contar primos menores que: **1000000**  
Hay 78498 primos menores que 1000000

En matemáticas, a este valor se le llama [función  \$\pi\$](#) .

7. Todos los números naturales mayores que 1 pueden ser factorizados de una única manera como un [producto de divisores primos](#).

Escriba un programa que muestre los factores primos de un número entero ingresado por el usuario:

```
Ingrese numero: 204
2
2
3
17
Ingrese numero: 8575
5
5
7
7
7
```

8. La [conjetura de Goldbach](#) sugiere que todo número par mayor que dos puede ser escrito como la suma de dos números primos. Hasta ahora no se conoce ningún número para el que esto no se cumpla.

Escriba un programa que reciba un número par como entrada y muestre todas las maneras en que puede ser escrito como una suma de dos primos:

Ingrese número par: **338**

7 + 331

31 + 307

61 + 277

67 + 271

97 + 241

109 + 229

127 + 211

139 + 199

157 + 181

Muestre sólo una de las maneras de escribir cada suma (por ejemplo, si muestra 61 + 271, no muestre 271 + 61).

9. Escriba programas que respondan las siguientes preguntas:

- o ¿Cuántos primos menores que diez mil terminan en 7?
- o ¿Cuál es la suma de los cuadrados de los números primos entre 1 y 1000? (Respuesta: 49.345.379).
- o ¿Cuál es el producto de todos los números primos menores que 100 que tienen algún dígito 7? (Respuesta:  $7 \times 17 \times 37 \times 47 \times 67 \times 71 \times 73 \times 79 \times 97 = 550.682.633.299.463$ ).

## 14- Máquina de alimentos

Una máquina de alimentos tiene productos de tres tipos, A, B y C, que valen respectivamente \$270, \$340 y \$390. La máquina acepta y da de vuelto monedas de \$10, \$50 y \$100.

Escriba un programa que pida al usuario elegir el producto y luego le pida ingresar las monedas hasta alcanzar el monto a pagar. Si el monto ingresado es mayor que el precio del producto, el programa debe entregar las monedas de vuelto, una por una.

Elija producto: **A**

Ingrese monedas:

**100**

**10**

**50**

**100**

**100**

Su vuelto:

**50**

**10**

**10**

**10**

**10**

Elija producto: **B**

Ingrese monedas:

**100**

**100**

**100**

**100**

Su vuelto:



50  
10

Elija producto: **C**

Ingrese monedas:

100  
100  
50  
10  
100  
10  
10  
10

## 15- Supermercado

Un supermercado utiliza tablas de datos para llevar la información de su inventario.

En un programa, cada tabla de datos es una lista de tuplas.

La lista `productos` tiene el código, el nombre, el precio y la cantidad de unidades del producto en bodega:

Código	Nombre	Precio (\$)	Stock (unidades)
41419	Fideos	18	210
70717	Cuaderno	20	119
78714	Jabon	11	708
30877	Desodorante	30	79
47470	Yogurt	15	832
50809	Palta	27	55
75466	Galletitas	13	0
33692	Bebida	19	20
89148	Arroz	23	121
66194	Lapiz,	12	900
15982	Lámpara	250	40
41235	Chocolate	45	48

La lista `clientes` tiene el cuit y el nombre de los clientes del supermercado:

CUIT	Razón Social/Nombre
20-11652624-7	Clark Kent
27-8830268-0	Diana Prince
30-7547896-8	Detective Comics SA

La lista `ventas` contiene las ventas realizadas, representadas por el número de boleta, la fecha de la venta y el rut del cliente:

Boleta	Fecha	CUIT
1	12/09/2014	27-8830268-0
2	19/09/2014	20-11652624-7
3	30/09/2014	30-7547896-8
4	01/10/2014	27-8830268-0
5	13/10/2014	30-7547896-8
6	11/11/2014	20-11652624-7

El detalle de cada venta se encuentra en la lista `items`. Cada ítem tiene asociado un número de boleta, un código de producto y una cantidad:

Boleta	Código Producto	Cantidad vendida
1	89148	3
2	50809	4
2	33692	2
2	47470	6
3	30877	1
4	89148	1
4	75466	2
5	89148	2
5	47470	10
6	41419	2

Por ejemplo, en la venta con boleta número 2, fueron vendidas 4 paltas, 2 bebidas y 6 yogures.

- Cree los arrays necesarios para almacenar la información brindada
- Escriba los siguientes métodos y sus llamadas para testearlos desde el Main del programa:
  - `ProductoMasCaro(productos)`  
Devuelve la descripción del producto más caro de la lista `productos`
  - `ValorTotalStock(productos)`  
Devuelve la suma total de los productos en stock
  - `TotalVentas(items, productos)`  
Devuelve la suma total de las ventas registradas en `items`
  - `ProductoMasVendido(items, productos)`  
Descripción del producto con mas ventas
  - `ProductoQueMasIngresosGenero(items, productos)`  
Descripción del producto que genero mas ingresos
  - `MejorCliente(items, productos, clientes)`  
Nombre/Razón Social del cliente que mas pago
  - `TotalVentasMes(2014, 10, items, productos)`

Sumatoria de las ventas realizadas en el mes indicado

- FechaUltimaVentaProducto(47470, items, ventas)  
Ultima fecha en que se registra venta para el producto indicado
- ListarVentas(27-8830268-0)  
Listar todas las ventas agrupadas por Cliente y boleta

Cliente	Boleta	Cod.Prod	Cantidad	Importe
Total				
Diana Prince (27-8830268-0)	1	89148	3	23
69				
Boleta 1 Total: 69				

Diana Prince (27-8830268-0)	4	89148	1	23
23				
Diana Prince (27-8830268-0)	4	75466	2	13
26				
Boleta 4 Total: 49				

Total Diana Prince (27-8830268-0): 78

**No preocuparse por el alineado del encabezado**

- ImprimirListado(2014,10)  
Imprimir el listado completo de ventas agrupado por clientes y boleta para un mes
- c. Ahora en lugar de arrays de tipos primitivos, defina las estructuras necesarias para almacenar la información de las listas (Cliente, Producto, Item, Venta).

Reescriba los metodos para utilizar las estructuras en lugar de los arrays